

AMENDMENTS TO THE CLAIMS

1. (Cancelled)

2. (Currently Amended) ~~A method for controlling upgrade of firmware of an electronic device according to claim 1~~ A method for controlling upgrade of firmware of an electronic device comprising process of:

forming a hardware ID code;

forming a firmware ID code; and

checking said firmware ID code and hardware ID code and upgrading said firmware,

wherein said process of forming a hardware ID code further comprises steps of:

confirming hardware version;

confirming a password;

confirming a salt;

confirming an algorithm;

operating said algorithm with said hardware version, password and salt and getting a hardware ID code; and

writing said salt and hardware ID code into said hardware.

3. (Currently Amended) ~~A method for controlling upgrade of firmware of an electronic device according to claim 1~~ A method for controlling upgrade of firmware of an electronic device comprising process of:

forming a hardware ID code;

forming a firmware ID code; and
checking said firmware ID code and hardware ID code and upgrading said firmware,
wherein said process of forming a firmware ID code further comprises steps of:
taking software version and hardware version;
computing and getting said hardware ID code and said salt;
computing and getting a checksum of firmware file; and
filling in empty spaces with random numbers.

4. (Currently Amended) ~~A method for controlling upgrade of firmware of an electronic device according to claim 1~~ A method for controlling upgrade of firmware of an electronic device comprising process of:

forming a hardware ID code;
forming a firmware ID code; and
checking said firmware ID code and hardware ID code and upgrading said firmware,
wherein said process of checking said firmware ID code and hardware ID code and upgrading said firmware further comprises steps of:
assuring system resources capable of storing a new firmware;
loading said new firmware;
confirming a checksum of said new firmware;
taking ID codes of said new firmware and an original salt stored in said hardware for
computing a hardware ID code;

checking if said computed hardware ID code is same as that stored in said product hardware; and

recording said new firmware into a flash memory according to said checking result, and restarting automatically.

5. (Currently Amended) A method for controlling upgrade of firmware of an electronic device according to claim 2 wherein said hardware version is in a format of: comprising:

Vender/Product ID	V	C	P	H	R
------------------------------	--------------	--------------	--------------	--------------	--------------

~~wherein a~~ Vender/Product ID relatedd to the vender name and the product name;

~~V~~ is a byte of vender code;

~~C~~ is a byte of CPU code;

~~P~~ is a byte of product code;

~~H~~ is a byte of hardware code; and

~~R~~ is a reserved byte for extension.

6. (Currently Amended) A method for controlling upgrade of firmware of an electronic device according to claim 3 wherein said hardware version is in a format of: comprising:

Vender/Product ID	V	C	P	H	R
------------------------------	--------------	--------------	--------------	--------------	--------------

~~wherein~~ Vender/Product ID relatedd to the vender name and the product name;

~~V~~ is a byte of vender code;

~~C~~ is a byte of CPU code;

~~P~~ is a byte of product code;

~~H~~ is a byte of hardware code; and

~~R~~ is a reserved byte for extension.

7. (Currently Amended) A method for controlling upgrade of firmware of an electronic device according to claim 3 wherein said software version is in a format of: comprising:

A	B	B	B	C	C
---	---	---	---	---	---

wherein A is a byte of main version code;

~~first B~~ is a code for major bug correction;

~~second and third B~~ are codes for secondary bug corrections; and

~~CC~~ are codes of specified version.

8. (Currently Amended) A method for controlling upgrade of firmware of an electronic device according to ~~claim 1~~ claim 2 wherein said firmware ID code is in a format of: comprising:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20h	Vendor/Product ID											V	C	P	H	R
30h	CS	Rand1	salt	H(version(20h-2Fh), key, salt(33h))												
40h	Software Dynamic Version							Rand2								

wherein Vendor/Product ID relates to the vender name and product name;

~~V~~ is a byte of vender code;

~~C~~ is a byte of CPU code;

~~P~~ is a byte of product code;

~~H~~ is a byte of hardware code;

~~R~~ is a reserved byte for extension;

~~CS~~ is a byte of checksum;

~~salt~~ is a byte of randomized number;

~~H()~~ is a byte of hardware ID code;

~~Software dynamic version~~ is a 6-byte software version code, and

~~Rand1 and Rand2~~ are randomized numbers for filling the empty spaces.

9. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 2 wherein said salt is randomly generated, and being different each time.

10. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 3 wherein said salt is randomly generated, and being different each time.

11. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 4 wherein said salt is randomly generated, and being different each time.

12. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 2 wherein said algorithm is logical exclusive OR operation.

13. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 3 wherein said computing is algorithm of logical exclusive OR operation.

14. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 4 wherein said computing is algorithm of logical exclusive OR operation.

15. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 2 wherein said algorithm is MD5 message-digest algorithm.

16. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 3 wherein said computing is algorithm of MD5 message-digest algorithm.

17. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 4 wherein said computing is algorithm of MD5 message-digest algorithm.

18. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 2 wherein said salt and said hardware ID code are both written into said device and stored in a non-evaporative memory unit of said hardware.

19. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 18 wherein said non-evaporative memory unit is a complex programmable logic device.

20. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 18 wherein said non-evaporative memory unit is an electrically erasable programmable read only memory.

21. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 4 wherein during said step of assuring system resources capable of storing a new firmware, a warning message is generated to inform user to restart when said system resources are insufficient.

22. (Original) A method for controlling upgrade of firmware of an electronic device according to claim 4 wherein during said steps of confirming and checking, a warning message is generated to inform user to restart when a checking result is not complied.